Risoluzione dei problemi relativi al PIX per il passaggio del traffico di dati su un tunnel IPSec stabilito

Sommario

Introduzione

Questo documento affronta e fornisce una soluzione al problema del perché un tunnel IPsec stabilito correttamente da un client VPN Cisco a un PIX non è in grado di passare i dati.

L'impossibilità di passare dati su un tunnel IPsec stabilito tra un client VPN e un PIX si verifica spesso quando non è possibile eseguire il ping o il trasferimento in modalità Telnet da un client VPN a nessuno degli host della LAN dietro il PIX. In altre parole, il client VPN e il PIX non possono passare dati crittografati tra loro. Questo si verifica perché il PIX ha un tunnel IPsec da LAN a LAN verso un router e anche un client VPN. L'impossibilità di passare i dati è il risultato di una configurazione con lo stesso elenco di controllo di accesso (ACL) sia per il nat 0 che per la mappa crittografica statica del peer IPsec da LAN a LAN.

Prerequisiti

Requisiti

Nessun requisito specifico previsto per questo documento.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Cisco Secure PIX Firewall 6.0.1
- Router Cisco 1720 con software Cisco IOS® versione 12.2(6)

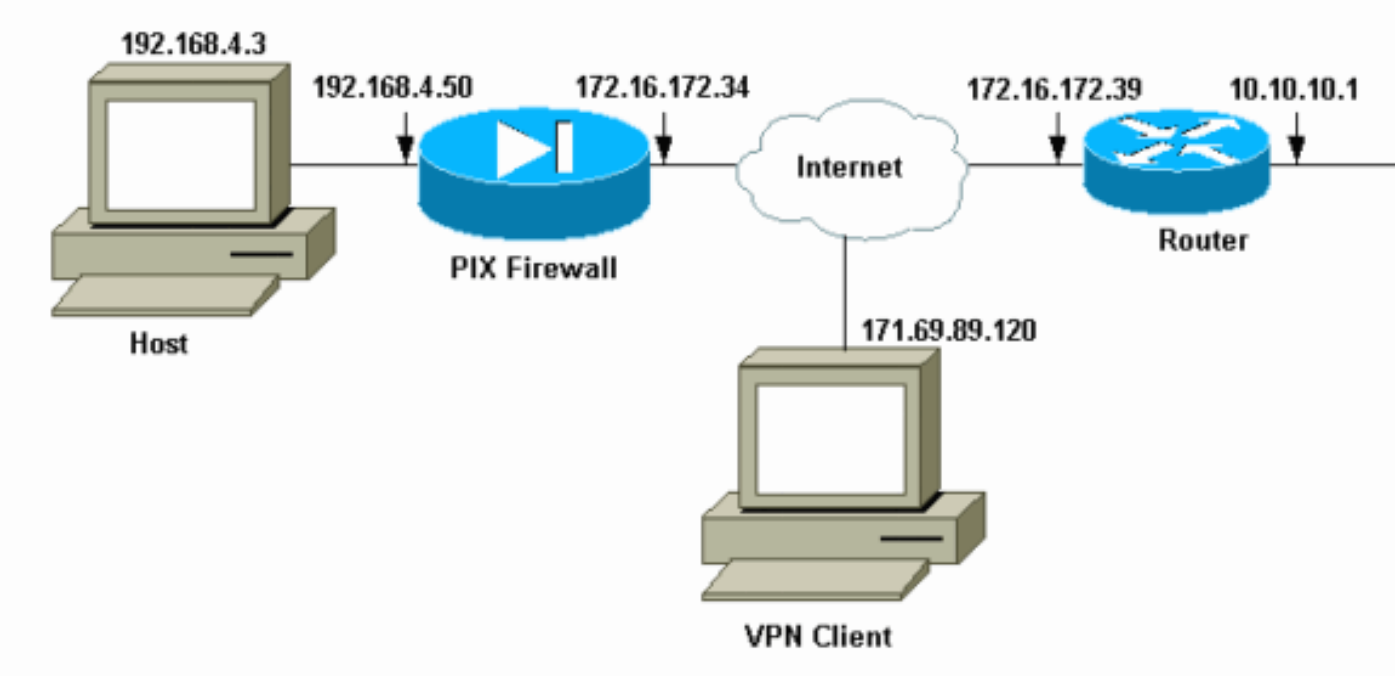
Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Fare riferimento a Cisco Technical Tips Conventions per ulteriori informazioni sulle convenzioni dei documenti.

Risoluzione dei problemi relativi a PIX

Esempio di rete



Configurazione di esempio con problemi

PIX 520
pix520-1#**write terminal** Building configuration... : Saved : PIX Version 6.0(1) nameif ethernet0 outside security0 nameif ethernet1 inside security100 enable password 2KFQnbNIdI.2KYOU encrypted passwd 2KFQnbNIdI.2KYOU encrypted hostname pix520-1

```
domain-name vpn.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
!--- Access-List "140" defines interesting traffic to
bypass NAT for VPN !--- and defines VPN interesting
traffic. This is incorrect. access-list 140 permit ip
192.168.4.0 255.255.255.0 10.10.10.0 255.255.255.0
access-list 140 permit ip 192.168.4.0 255.255.255.0
10.1.2.0 255.255.255.0
no pager
logging on
logging console debugging
logging monitor debugging
logging buffered debugging
logging trap debugging
logging history debugging
logging host outside 192.168.2.6
interface ethernet0 auto
interface ethernet1 auto
mtu outside 1500
mtu inside 1500
!--- IP addresses on the outside and inside interfaces.
ip address outside 172.16.172.34 255.255.255.240
ip address inside 192.168.4.50 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
ip local pool ippool 10.1.2.1-10.1.2.254
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
pdm history enable
arp timeout 14400
global (outside) 1 172.16.172.57 netmask 255.255.255.255
!--- The nat 0 command bypasses NAT for the packets
destined over the IPsec tunnel.

Nat (inside) 0 access-list 140
Nat (inside) 1 0.0.0.0 0.0.0.0 0 0
route outside 0.0.0.0 0.0.0.0 172.16.172.33 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc
0:10:00 h323
0:05:00 sip
0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
AAA-server RADIUS protocol radius
AAA-server mytest protocol tacacs+
AAA-server nasir protocol radius
snmp-server host outside 192.168.2.6
no snmp-server location
no snmp-server contact
snmp-server community public
snmp-server enable traps
floodguard enable
```

```
!--- The sysopt command bypasses conduits or ACLs that
check to be applied !--- on the inbound VPN packets
after decryption.

sysopt connection permit-ipsec
no sysopt route dnat
!--- The crypto ipsec command defines IPsec encryption
and authen algo.

crypto ipsec transform-set myset esp-des esp-md5-hmac
crypto dynamic-map dynmap 10 set transform-set myset
!--- The crypto map commands define the IPsec !---
Security Assocation (SA) (Phase II SA) parameters.

crypto map mymap 5 ipsec-isakmp
crypto map mymap 5 match address 140
crypto map mymap 5 set peer 172.16.172.39
crypto map mymap 5 set transform-set myset
crypto map mymap 10 ipsec-isakmp dynamic dynmap
crypto map mymap interface outside
isakmp enable outside
!--- The isakmp key command defines the pre-shared key
for the peer address.

isakmp key ******** address 172.16.172.39 netmask
255.255.255.255 no-xauth
no-config-mode
isakmp identity address
!--- The isakmp policy defines the Phase 1 SA
parameters.

isakmp policy 10 authentication pre-share
isakmp policy 10 encryption des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption Des
isakmp policy 20 hash sha
isakmp policy 20 group 1
isakmp policy 20 lifetime 86400
vpngroup vpn3000 address-pool ippool
vpngroup vpn3000 idle-time 1800
vpngroup vpn3000 password ********
telnet 192.168.4.0 255.255.255.0 inside
telnet 171.69.89.82 255.255.255.255 inside
telnet timeout 5
ssh 172.0.0.0 255.0.0.0 outside
ssh 171.0.0.0 255.255.255.0 outside
ssh 171.0.0.0 255.0.0.0 outside
ssh timeout 60
terminal width 80
Cryptochecksum:55948dc706cc700e9c10e1d24a8b125c
```

In una configurazione con problemi, il traffico interessante, o il traffico da crittografare per il tunnel da LAN a LAN, è definito da ACL 140. La configurazione usa lo stesso ACL dell'ACL nat 0.

Comprendere la sequenza generale di eventi

Quando un pacchetto IP arriva all'interfaccia interna del PIX, viene controllato Network Address Translation (NAT). Quindi, vengono controllati gli ACL per le mappe crittografiche.

- **Modalità di utilizzo di nat 0.**L'ACL nat 0 definisce ciò che non deve essere incluso in NAT. L'ACL nel comando **nat 0** definisce l'indirizzo di origine e di destinazione per cui le regole NAT sul PIX sono disabilitate. Pertanto, un pacchetto IP con indirizzo di origine e destinazione corrispondente all'ACL definito nel comando **nat 0** ignora tutte le regole NAT sul PIX.Per implementare i tunnel da LAN a LAN tra un PIX e un altro dispositivo VPN con l'aiuto degli indirizzi privati, usare il comando **nat 0** per ignorare NAT. Le regole sul firewall PIX impediscono che gli indirizzi privati vengano inclusi in NAT mentre queste regole vanno alla LAN remota sul tunnel IPsec.
- **Come viene usato l'ACL crittografico.**Dopo le ispezioni NAT, il PIX controlla l'origine e la destinazione di ciascun pacchetto IP che arriva alla sua interfaccia interna in modo da farli corrispondere agli ACL definiti nelle mappe crittografiche statiche e dinamiche. Se il PIX rileva una corrispondenza con l'ACL, effettua una delle seguenti operazioni:Se non esiste alcuna associazione di sicurezza (SA) IPSec corrente già creata con il dispositivo IPSec peer per il traffico, il PIX avvia le negoziazioni IPSec. Dopo aver generato le associazioni di protezione, il pacchetto viene crittografato e inviato al peer IPsec tramite il tunnel IPsec.Se esiste già un'associazione di protezione IPsec creata con il peer, il PIX crittografa il pacchetto IP e lo invia al dispositivo IPsec del peer.
- **ACL dinamico.**Quando un client VPN si connette al PIX con l'aiuto di IPsec, il PIX crea un ACL dinamico che specifica l'indirizzo di origine e di destinazione da utilizzare per definire il traffico interessante per questa connessione IPsec.

Comprendere la serie problematica di eventi sul PIX

Un errore comune nella configurazione è quello di usare lo stesso ACL per nat 0 e le mappe crittografiche statiche. In queste sezioni vengono illustrati i motivi per cui tale operazione genera un errore e viene illustrato come correggere il problema.

La configurazione PIX mostra che l'ACL 140 nat 0 ignora NAT quando i pacchetti IP passano dalla rete 192.168.4.0/24 alle reti 10.10.10.0/24 e 10.1.2.0/24 (indirizzo di rete definito nel pool locale IP). Inoltre, l'ACL 140 definisce il traffico interessante per la mappa crittografica statica del peer 172.16.172.39.

Quando un pacchetto IP arriva all'interfaccia PIX interna, il controllo NAT viene completato e quindi il PIX controlla gli ACL nelle mappe crittografiche. Il PIX inizia con la mappa crittografica con il numero di istanza più basso. Infatti la mappa crittografica statica dell'esempio precedente ha il numero di istanza più basso e quindi viene controllato l'ACL 140. Quindi, viene controllato l'ACL dinamico per la mappa crittografica dinamica. In questa configurazione, l'ACL 140 è definito per crittografare il traffico proveniente dalla rete 192.168.4.0 /24 e diretto alle reti 10.10.10.0/24.0 e 10.1.2.0 /24. Tuttavia, per il tunnel LAN-LAN, è necessario crittografare solo il traffico tra le reti 192.168.4.0 /24 e 10.10.10.0 /24. In questo modo il router peer IPsec definisce il proprio ACL crittografico.

Comprendere la serie problematica di eventi sul PIX

Quando un client stabilisce una connessione IPsec al PIX, gli viene assegnato un indirizzo IP dal pool locale IP. In questa istanza, al client viene assegnata la versione 10.1.2.1. Il PIX genera anche un ACL dinamico, come mostrato nell'output del comando **show crypto map**:

```
Crypto Map "mymap" 20 ipsec-isakmp
```

```
Peer = 171.69.89.120
access-list dynacl2 permit ip host 172.16.172.34 host 10.1.2.1 (hitcnt=0)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
Crypto Map "mymap" 30 ipsec-isakmp
Peer = 171.69.89.120
access-list dynacl3 permit ip any host 10.1.2.1 (hitcnt=0)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
pix520-1(config)#
```

Il comando **show crypto map** mostra anche la mappa crittografica statica:

```
Crypto Map: "mymap" interfaces: { outside }
Crypto Map "mymap" 5 ipsec-isakmp
Peer = 172.16.172.39
access-list 140 permit ip 192.168.4.0 255.255.255.0 10.10.10.0255.255.255.0
   (hitcnt=45)
access-list 140 permit ip 192.168.4.0 255.255.255.0 10.1.2.0 255.255.255.0
   (hitcnt=84)
Current peer: 172.16.172.39
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset,}
```

Una volta stabilito il tunnel IPsec tra il client e il PIX, il client avvia un ping verso l'host 192.168.4.3. Quando riceve la richiesta echo, l'host 192.168.4.3 risponde con una risposta echo come mostrato in questo output del comando **debug icmp trace**.

```
27: Inbound ICMP echo request (len 32 id 2 seq 7680)
   10.1.2.1 > 192.168.4.3> 192.168.4.3
28: Outbound ICMP echo reply (Len 32 id 2 seq 7680)
   192.168.4.3 >192.168.4.3 > 10.1.2.1
29: Inbound ICMP echo request (Len 32 id 2 seq 7936)
   10.1.2.1 > 192.168.4.3> 192.168.4.3
30: Outbound ICMP echo reply (Len 32 id 2 seq 7936)
   192.168.4.3 >192.168.4.3 > 10.1.2.1
```

Tuttavia, la risposta echo non raggiunge il client VPN (host 10.1.2.1) e il ping ha esito negativo. Per visualizzare questo messaggio, usare il comando **show crypto ipsec sa** sul PIX. Questo output mostra che il PIX decrittografa 120 pacchetti provenienti dal client VPN, ma non crittografa alcun pacchetto né invia i pacchetti crittografati al client. Pertanto, il numero di pacchetti incapsulati è zero.

```
pix520-1(config)#show crypto ipsec sa
interface: outside
Crypto map tag: mymap, local addr. 172.16.172.34
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (10.1.2.1/255.255.255.255/0/0)
current_peer: 171.69.89.120
dynamic allocated peer ip: 10.1.2.1
PERMIT, flags={}
```

```
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
!--- No packets encrypted and sent to client. #pkts decaps: 120, #pkts decrypt: 120, #pkts
verify 120
!--- 120 packets received from client. #pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
local crypto endpt.: 172.16.172.34, remote crypto endpt.: 171.69.89.120
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 33a45029
inbound esp sas:
spi: 0x279fc5e9(664782313)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 5, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607985/27809)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound pcp sas:
outbound ESP sas:
spi: 0x33a45029(866406441)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 6, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4608000/27809)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
local ident (addr/mask/prot/port): (192.168.4.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.10.10.0/255.255.255.0/0/0)
current_peer: 172.16.172.39
PERMIT, flags={origin_is_acl,}
#pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
#pkts decaps: 23, #pkts decrypt: 23, #pkts verify 23
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. Failed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
local crypto endpt.: 172.16.172.34, remote crypto endpt.: 172.16.172.39
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: f264e92c
inbound ESP sas:
spi: 0x2772b869(661829737)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 1, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607997/2420)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound PCP sas:
outbound ESP sas:
spi: 0xf264e92c(4066699564)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607999/2420)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
```

Nota: quando l'host 192.168.4.3 risponde alla richiesta echo, il pacchetto IP arriva all'interfaccia interna del PIX.

```
38: Outbound ICMP echo reply (Len 32 id 2 seq 8960)
   192.168.4.3 >192.168.4.3 > 10.1.2.1
```

Una volta che il pacchetto IP arriva all'interfaccia interna, il PIX controlla l'ACL 140 nat 0 e determina che gli indirizzi di origine e di destinazione del pacchetto IP corrispondano all'ACL. Pertanto, questo pacchetto IP ignora tutte le regole NAT sul PIX. Quindi, controllare gli ACL crittografici. Poiché la mappa crittografica statica ha il numero di istanza più basso, il relativo ACL viene controllato per primo. Poiché in questo esempio viene usato ACL 140 per la mappa crittografica statica, il PIX controlla questo ACL. Ora, il pacchetto IP ha un indirizzo di origine di 192.168.4.3 e una destinazione di 10.1.2.1. Poiché questo indirizzo corrisponde all'ACL 140, il PIX ritiene che il pacchetto IP sia destinato al tunnel IPsec da LAN a LAN con peer 172.16.172.39 (a differenza dei nostri obiettivi). Pertanto, controlla il database SA per verificare se esiste già un'associazione di sicurezza corrente con peer 172.16.72.39 per questo traffico. Come mostrato nell'output del comando **show crypto ipsec sa**, non esiste alcuna associazione di protezione per questo traffico. Il PIX non cripta né invia il pacchetto al client VPN. Al contrario, avvia un'altra negoziazione IPsec con il peer 172.16.172.39, come mostrato nell'output:

```
crypto_isakmp_process_block: src 172.16.172.39, dest 172.16.172.34
return status is IKMP_NO_ERR_NO_TRANS02303: sa_request, (key eng. msg.)
src= 172.16.172.34, dest= 172.16.172.39,
src_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
dest_proxy= 10.1.2.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform=
ESP-Des esp-md5-hmac , lifedur= 28800s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
702303: sa_request, (key Eng. msg.) src= 172.16.172.34, dest=
172.16.172.39, src_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
dest_proxy= 10.1.2.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform=
ESP-Des esp-md5-hmac , lifedur= 28800s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
ISAKMP (0): sending NOTIFY message 36137 protocol 1
return status is IKMP_NO_ERR_NO_TRANSIPSEC(key_engine): request timer
fired: count = 2,
(identity) local= 172.16.172.34, remote= 172.16.172.39,
local_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
remote_proxy= 10.1.2.0/255.255.255.0/0/0 (type=4)
```

La negoziazione IPSec non ha esito positivo per i seguenti motivi:

- La versione 172.16.172.39 definisce solo le reti 10.10.10.0/24 e 192.168.4.0/24 come il traffico interessante nel proprio ACL per la mappa crittografica peer 172.16.172.34.
- Le identità proxy non corrispondono durante la negoziazione IPSec tra i due peer.
- Se il peer avvia la negoziazione e la configurazione locale specifica PFS (Perfect Forward Secrecy), deve eseguire uno scambio PFS oppure la negoziazione non riesce. Se la configurazione locale non specifica un gruppo, viene utilizzato il valore predefinito group1 e viene accettata un'offerta di group1 o group2. Se nella configurazione locale è specificato il gruppo 2, tale gruppo deve far parte dell'offerta del peer, altrimenti la negoziazione avrà esito negativo. Se la configurazione locale non specifica PFS, accetta qualsiasi offerta di PFS dal peer. Il gruppo del modulo principale Diffie-Hellman a 1024 bit, group2, offre una maggiore protezione rispetto a group1, ma richiede un tempo di elaborazione maggiore rispetto a group1.**Nota:** il comando **crypto map set pfs** imposta IPsec in modo che richieda PFS quando vengono richieste nuove associazioni di protezione per questa voce della mappa crittografica. Utilizzare il comando **no crypto map set pfs** per specificare che IPsec non richiede PFS. Questo comando è disponibile solo per le voci della mappa crittografica IPsec-ISAKMP e per le voci della mappa crittografica dinamica. Per impostazione predefinita, PFS non è richiesto.

Con PFS, ogni volta che viene negoziata una nuova associazione di protezione si verifica un nuovo scambio Diffie-Hellman. Ciò richiede tempi di elaborazione aggiuntivi. PFS aggiunge un altro livello di protezione perché se una chiave viene danneggiata da un utente non autorizzato, vengono compromessi solo i dati inviati con tale chiave. Durante la negoziazione, questo comando determina la richiesta di IPsec a PFS quando vengono richieste nuove SA per la voce della mappa crittografica. L'impostazione predefinita (group1) viene inviata se l'istruzione **set pfs** non specifica un gruppo.**Nota:** le negoziazioni IKE con un peer remoto possono bloccarsi quando un firewall PIX dispone di numerosi tunnel provenienti dal firewall PIX e terminanti su un singolo peer remoto. Questo problema si verifica quando PFS non è abilitato e il peer locale richiede molte richieste di reimpostazione delle chiavi simultanee. Se il problema si verifica, l'associazione di protezione IKE non viene ripristinata finché non scade o finché non viene cancellata manualmente con il comando **clear [crypto] isakmp sa**. Le unità del firewall PIX configurate con molti tunnel per molti peer o molti client che condividono lo stesso tunnel non sono interessate da questo problema. Se la configurazione è interessata, abilitare PFS con il comando **crypto map mapname seqnum set pfs**.

I pacchetti IP sul PIX vengono infine scartati.

Comprendere la soluzione

Per correggere l'errore, definire due ACL separati per nat 0 e le mappe crittografiche statiche. A tale scopo, l'esempio definisce ACL 190 per il comando **nat 0** e usa l'ACL 140 modificato per la mappa crittografica statica, come mostrato nell'output.

PIX 520-1

```
pix520-1(config)#
pix520-1(config)#write terminal
Building configuration...
: Saved
:
PIX Version 6.0(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 2KFQnbNIdI.2KYOU encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pix520-1
domain-name vpn.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
!--- Access list 140 defines interesting traffic in
order to bypass NAT for VPN. access-list 140 permit ip
192.168.4.0 255.255.255.0 10.10.10.0255.255.255.0
!--- Defines VPN interesting traffic. access-list 190
permit ip 192.168.4.0 255.255.255.0
10.10.10.0255.255.255.0
access-list 190 permit ip 192.168.4.0 255.255.255.0
10.1.2.0 255.255.255.0
no pager
```

```
logging on
logging console debugging
logging monitor debugging
logging buffered debugging
logging trap debugging

logging history debugging
logging host outside 192.168.2.6
interface ethernet0 auto
interface ethernet1 auto
mtu outside 1500
mtu inside 1500
ip address outside 172.16.172.34 255.255.255.240
ip address inside 192.168.4.50 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
ip local pool ippool 10.1.2.1-10.1.2.254
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
pdm history enable
arp timeout 14400
global (outside) 1 172.16.172.57 netmask 255.255.255.255
!--- The nat 0 command bypasses NAT for the packets
destined over the IPsec tunnel..

Nat (inside) 0 access-list 190
Nat (inside) 1 0.0.0.0 0.0.0.0 0 0
route outside 0.0.0.0 0.0.0.0 172.16.172.33 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc
0:10:00 h323
0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
AAA-server TACACS+ protocol tacacs+
AAA-server RADIUS protocol radius
AAA-server mytest protocol tacacs+
AAA-server nasir protocol radius
snmp-server host outside 192.168.2.6
no snmp-server location
no snmp-server contact
snmp-server community public
snmp-server enable traps
floodguard enable
sysopt connection permit-ipsec
no sysopt route dnat
crypto ipsec transform-set myset ESP-Des esp-md5-hmac
crypto dynamic-map dynmap 10 set transform-set myset
!--- The crypto map commands define the IPsec SA (Phase
II SA) parameters.

crypto map mymap 5 ipsec-isakmp
crypto map mymap 5 match address 140
crypto map mymap 5 set peer 172.16.172.39
crypto map mymap 5 set transform-set myset
crypto map mymap 10 ipsec-isakmp dynamic dynmap
crypto map mymap interface outside
isakmp enable outside
isakmp key ******** address 172.16.172.39 netmask
255.255.255.255 no-xauth
no-config-mode
isakmp identity address
```

```
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption Des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption Des
isakmp policy 20 hash sha
isakmp policy 20 group 1
isakmp policy 20 lifetime 86400
vpngroup vpn3000 address-pool ippool
vpngroup vpn3000 idle-time 1800
vpngroup vpn3000 password ********
telnet 192.168.4.0 255.255.255.0 inside
telnet 171.69.89.82 255.255.255.255 inside
telnet timeout 5
ssh 172.0.0.0 255.0.0.0 outside
ssh 171.0.0.0 255.255.255.0 outside
ssh 171.0.0.0 255.0.0.0 outside
ssh timeout 60
terminal width 80
Cryptochecksum:e2cb98b30d3899597b3af484fae4f9ae
: end
[OK]
pix520-1(config)# pix520-1(config)#show crypto map
```

Dopo aver apportato le modifiche e aver stabilito un tunnel IPsec con il PIX, il client usa il comando show crypto map. Questo comando mostra che per la mappa crittografica statica, il traffico interessante definito da ACL 140 è solo 192.168.4.0/24 e 10.10.10.0/24, ossia l'obiettivo originale. Inoltre, l'elenco degli accessi dinamici mostra il traffico interessante definito come client (10.1.2.1) e PIX (172.16.172.34).

```
pix520-1(config)#show crypto map
Crypto Map: "mymap" interfaces: { outside }
Crypto Map "mymap" 5 ipsec-isakmp
Peer = 172.16.172.39
access-list 140 permit ip 192.168.4.0 255.255.255.0 10.10.10.0255.255.255.0
   (hitcnt=57)
Current peer: 172.16.172.39
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
Crypto Map "mymap" 10 ipsec-isakmp
Dynamic map template tag: dynmap
Crypto Map "mymap" 20 ipsec-isakmp
Peer = 171.69.89.120
access-list dynacl4 permit ip host 172.16.172.34 host 10.1.2.1 (hitcnt=0)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
Crypto Map "mymap" 30 ipsec-isakmp
Peer = 171.69.89.120
access-list dynacl5 permit ip any host 10.1.2.1 (hitcnt=13)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
```

Quando il client VPN 10.1.2.1 invia un ping all'host 192.168.4.3, la risposta echo arriva

all'interfaccia interna del PIX. Il PIX controlla l'ACL 190 nat 0 e determina che il pacchetto IP corrisponde all'ACL. Pertanto, il pacchetto ignora le regole NAT sul PIX. Quindi, il PIX controlla la mappa crittografica statica ACL 140 per trovare una corrispondenza. Questa volta, l'origine e la destinazione del pacchetto IP non corrispondono all'ACL 140. Pertanto, il PIX controlla l'ACL dinamico e trova una corrispondenza. Il PIX controlla quindi il proprio database SA per verificare se è già stata stabilita un'associazione di protezione IPsec con il client. Poiché il client ha già stabilito una connessione IPSec con il PIX, esiste un'associazione di protezione IPSec. Il PIX cripta quindi i pacchetti e li invia al client VPN. Usare l'output del comando **show crypto ipsec sa** restituito dal PIX per verificare che i pacchetti siano entrambi crittografati e decriptati. In questo caso, il PIX criptò sedici pacchetti e li inviò al client. Il PIX ha anche ricevuto pacchetti crittografati dal client VPN e decriptato sedici pacchetti.

```
pix520-1(config)#show crypto ipsec sa
interface: outside
Crypto map tag: mymap, local addr. 172.16.172.34
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (10.1.2.1/255.255.255.255/0/0)
current_peer: 171.69.89.120
dynamic allocated peer ip: 10.1.2.1
PERMIT, flags={}
#pkts encaps: 16, #pkts encrypt: 16,#pkts digest 16
#pkts decaps: 16, #pkts decrypt: 16, #pkts verify 16
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. Failed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
local crypto endpt.: 172.16.172.34, remote crypto endpt.: 171.69.89.120
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 613d083d
inbound ESP sas:
spi: 0x6adf97df(1793038303)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 4, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607998/27420)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound PCP sas:
outbound ESP sas:
spi: 0x613d083d(1631389757)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 3, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607999/27420)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
local ident (addr/mask/prot/port): (192.168.4.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.10.10.0/255.255.255.0/0/0)
current_peer: 172.16.172.39
PERMIT, flags={origin_is_acl,}
#pkts encaps: 9, #pkts encrypt: 9, #pkts digest 9
#pkts decaps: 9, #pkts decrypt: 9, #pkts verify 9
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. Failed: 0, #pkts decompress failed: 0
#send errors 1, #recv errors 0
local crypto endpt.: 172.16.172.34, remote crypto endpt.: 172.16.172.39
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 58009c01
```

```
inbound ESP sas:
spi: 0x2d408709(759203593)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607998/3319)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound PCP sas: outbound ESP sas:
spi: 0x58009c01(1476434945)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 1, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607999/3319)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
pix520-1(config)# sh cr isa sa
Total : 2
Embryonic : 0
dst src state pending created
172.16.172.39 172.16.172.34 QM_IDLE 0 1
172.16.172.34 171.69.89.120 QM_IDLE 0 2
pix520-1(config)# sh cr ipsec sa
```

Configurazione del router e output del comando show

Cisco 1720-1

```
1720-1#show run
Building configuration...
Current configuration : 1592 bytes
!
! Last configuration change at 21:08:49 PST Mon Jan 7
2002
! NVRAM config last updated at 18:18:17 PST Mon Jan 7
2002
!
version 12.2
no parser cache
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 1720-1
!
no logging buffered
enable secret 5 $1$6jAs$tNxI1a/2DYFAtPLyCDXjo/
enable password ww
!
username cisco password 0 cisco
memory-size iomem 15
clock timezone PST -8
ip subnet-zero
no ip domain-lookup
ip domain-name cisco.com
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
```

```
!
!--- The crypto isakmp policy command defines the Phase
1 SA parameters.

crypto isakmp policy 15
authentication pre-share
crypto isakmp key cisco123 address 172.16.172.34
!
!
!--- The crypto ipsec transform-set command defines
IPsec encryption !--- and authentication algorithims.

crypto ipsec transform-set myset ESP-Des esp-md5-hmac
!
!
!--- The crypto map command defines the IPsec SA (Phase
II SA) parameters..

crypto map vpn 10 ipsec-isakmp
set peer 172.16.172.34
set transform-set myset
match address 150
!
!
!
!
!
interface FastEthernet0
ip address 172.16.172.39 255.255.255.240
speed auto
!--- The crypto map applied to the outbound interface.
crypto map vpn
interface Ethernet0
ip address 10.10.10.1 255.255.255.240
speed auto
no ip route-cache
no ip mroute-cache
!
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.172.33
no ip http server
ip pim bidir-enable
!
!--- Access-list defines interesting VPN traffic.
access-list 150 permit ip 10.10.10.0 0.0.0.255
192.168.4.0 0.0.0.255
!
line con 0
line aux 0
line vty 0 4
exec-timeout 0 0
password cisco
no login
line vty 5 15
login
!
no scheduler allocate
end
1720-1#
```

```
1720-1#show crypto isa sa
DST src state conn-id slot
```

```
172.16.172.39 172.16.172.34 QM_IDLE 132 0
1720-1#show crypto ipsec sa
interface: FastEthernet0
Crypto map tag: vpn, local addr. 172.16.172.39
local ident (addr/mask/prot/port): (10.10.10.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (192.168.4.0/255.255.255.0/0/0)
current_peer: 172.16.172.34
PERMIT, flags={origin_is_acl,}
#pkts encaps: 9 #pkts encrypt: 9 #pkts digest 9
#pkts decaps: 9, #pkts decrypt: 9, #pkts verify 9
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. Failed: 0, #pkts decompress failed: 0
#send errors 7, #recv errors 0
local crypto endpt.: 172.16.172.39, remote crypto endpt.: 172.16.172.34
path mtu 1500, media mtu 1500
current outbound spi: 2D408709
inbound ESP sas:
spi: 0x58009C01(1476434945)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
!--- IPsec SA 200 as seen in the show crypto engine connection active command.

slot: 0, conn id: 200, flow_id: 1, crypto map: vpn
sa timing: remaining key lifetime (k/sec): (4607998/3144)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound PCP sas:
outbound ESP sas:
spi: 0x2D408709(759203593)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
!--- IPsec SA 201 as seen in the show crypto engine connection active command.

slot: 0, conn id: 201, flow_id: 2, crypto map: vpn
sa timing: remaining key lifetime (k/sec): (4607998/3144)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
1720-1#

1720-1#show crypto map
Interfaces using crypto map mymap:
Crypto Map "vpn" 10 ipsec-isakmp
Peer = 172.16.172.34
Extended IP access list 150
access-list 150 permit ip 10.10.10.0 0.0.0.255 192.168.4.0 0.0.0.255
Current peer: 172.16.172.34
Security association lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform sets={ myset, }
Interfaces using crypto map vpn: FastEthernet0
```

Informazioni correlate

- Software Cisco PIX Firewall
- Riferimenti per i comandi di Cisco Secure PIX Firewall
- Avvisi sui prodotti per la sicurezza (inclusi PIX)
- RFC (Requests for Comments)

- Negoziazione IPSec/protocolli IKE
- Documentazione e supporto tecnico – Cisco Systems